Bitonic st-orderings of biconnected planar graphs

Martin Gronemann

Institut für Informatik, Universität zu Köln, Germany
gronemann@informatik.uni-koeln.de

Abstract

Vertex orderings play an important role in the design of graph drawing algorithms. Compared to canonical orderings, *st*-orderings lack a certain property that is required by many drawing methods. In this paper, we propose a new type of *st*-ordering for biconnected planar graphs that relates the ordering to the embedding. We describe a linear-time algorithm to obtain such an ordering and demonstrate its capabilities with two applications.

1 Introduction

Being a fundamental part of incremental drawing procedures, various types of orderings have been developed and improved over the years. De Fraysseix, Pach and Pollack [3] introduced the canonical ordering to create straight-line drawings of maximal planar graphs. Afterwards, Kant [10] extended this concept to the triconnected case. Obtainable in linear-time, both have been used in the graph drawing literature extensively. A few attempts have been made to generalize them to the biconnected case by relaxing their properties [7, 9]. However, an alternative that in nature works for biconnected graphs and that can be computed in linear time, are *st*-orderings [6]. In the field of graph drawing, they have been used in several methods, reaching from the construction of visibility representations to drawings of non-planar graphs, see e.g. [4, 12]. Although canonical and *st*-orderings share some properties in the planar case, it seems that they are usually not used in the same context.

In the following, we investigate these differences in more detail, especially one property of canonical orderings that is used implicitly in many drawing algorithms. Consider the successors of a single vertex in the clockwise ordering as implied by the embedding. Then their ranks in the canonical ordering form an increasing and then decreasing sequence, i.e., a *bitonic* sequence. Common *st*-orderings do not necessarily have this property, rendering them unsuitable for some applications.

We counteract by introducing a new type of *st*-ordering for biconnected planar graphs: the *bitonic st-ordering*, an *st*-ordering in which the successors of every vertex appear in the aforementioned pattern. We show that every biconnected planar graph admits such an ordering. The proof is constructive and yields a linear-time algorithm that computes the ordering and a corresponding embedding. For the case where a fixed embedding is given, we prove that one cannot always find a bitonic *st*-order. In order to further support our idea, we briefly describe two applications. In the first one, we extend the straight-line algorithm of de Fraysseix, Pach and Pollack [3] to bitonic *st*-orderings. In the second one, we describe how to obtain a special visibility representation and then transform it into a rectilinear T-shaped polygon contact representation.

2 Preliminaries

In the following, we first introduce some notations and definitions that are used throughout this work. If not stated otherwise, we consider only simple, planar biconnected graphs. One exception is the following definition of *st*-orderings that does not require planarity.

▶ **Definition 1.** *Let $G = (V, E)$ be a biconnected graph with $s, t \in V$, $s \neq t$ and let $\pi : V \to \{1, \dots, |V|\}$ be the rank of the vertices in an ordering $s = v_1, v_2, \dots, v_n = t$, i.e., $\pi(v_i) = i$ with $1 \leq i \leq n$. π is called an st-ordering, if for all vertices $v \in V$ with $1 < \pi(v) < n$, there exists $(u, v), (v, w) \in E$ with $\pi(u) < \pi(v) < \pi(w)$.*

From now on we assume that a graph is planar and a corresponding combinatorial embedding is given. In that case an *st*-ordering π of G has a nice property which has been used in the graph drawing literature extensively [4]: When considering the circular order induced by the embedding, the set of predecessors and successors form a consecutive sequence in the circular order of the embedding at a vertex. We denote this ordered sequence of successors of a vertex v by $S(v) = \{w_1, \dots, w_m\}$ such that for $1 \leq i < m$, w_i precedes w_{i+1} in the circular clockwise order around v and $\pi(v) < \pi(w_i)$ holds for all $1 \leq i \leq m$. This property is particularly useful in an incremental drawing procedure. However, one has no control over which successor is placed when.

Consider a simple example where a vertex v has been placed that has three successors, let us say $S(v) = \{w_1, w_2, w_3\}$. Then, π may be chosen such that w_2 must be placed before w_1 and w_3, i.e., $\pi(w_2) < \pi(w_1)$ and $\pi(w_2) < \pi(w_3)$. This may cause problems when attaching the edges (v, w_1) and (v, w_3), since (v, w_2) has already been attached. This lack of control is avoided by the canonical ordering that is limited to triconnected planar graphs:

▶ **Definition 2** (Kant [10]). *Let $G = (V, E)$ be a triconnected plane graph and (v_1, v_2) an edge on the outer face. Let $V_1 \cup \dots \cup V_K$ be an ordered partition of V and G_k ($1 \leq k \leq K$) the subgraph induced by $V_1 \cup \dots \cup V_k$ with outer face C_k. $V_1 \cup \dots \cup V_K$ is a canonical ordering of G if:*

- *$V_1 = \{v_1, v_2\}$ and $V_K = \{v_n\}$, where v_n lies on the outer face and is a is a neighbor of v_1.*
- *Each C_k ($k > 1$) is a cycle containing (v_1, v_2).*
- *Each G_k is biconnected and internally triconnected.*
- *For $1 < k < K$ one of the two following conditions holds:*
  1. *$V_k = \{z\}$ is a singleton where z belongs to C_k and has at least one neighbor in $G - G_k$.*
  2. *$V_k = \{z_1, \dots, z_m\}$ where each z_i ($1 \leq i \leq m$) has at least one neighbor in $G - G_k$, and where z_1 and z_m each have one neighbor in C_{k-1}, and these are the only two neighbors of V_k in G_{k-1}.*

Clearly, a situation as in the small example cannot occur with canonical orderings, because of the biconnectivity of G_k. In fact one can go one step further and claim (as

	biconnected	successor	bitonic
st-ordering	yes	yes	no
biconnected shelling- & canonical ordering	yes	no	yes
canonical ordering	no	yes	yes
bitonic *st*-ordering	yes	yes	yes

Table 1. Comparison of the features of various orderings.

we did in the introduction) that the partition indices of the successors when considered in the clockwise ordering as implied by the embedding, form an increasing and then decreasing sequence. We will prove this for canonical orderings as an intermediate step in the main section of this paper. For now we refer to this as the bitonic property.

The concept of canonical ordering has been generalized to the biconnected case. Gutwenger and Mutzel [7] use an ordered partition of the vertices, referred to as *biconnected shelling order*, to create poly-line drawings in an incremental manner. A similar but more vertex ordering-based concept is used by Harel and Sardas [9]. They introduce the so called *biconnected canonical ordering* for drawing planar graphs in a straight-line style. In both definitions, the constraints of the triconnected version have been relaxed. But this generalization sacrifices an important property that is required for some applications. In the triconnected case, every vertex $v \in V_k$, except for $k = K$, has a neighbor in $G - G_k$. We are not aware of any canonical ordering-like approach for the biconnected case, where this is guaranteed. In order to draw a connection to *st*-orderings, we refer to this property as the successor property. Table 1 summarizes the orderings and their features including our contribution (*bitonic st-ordering*).

Another common technique for the biconnected case that can be found in the literature is to first develop an algorithm using the canonical ordering and is therefore limited to triconnected graphs. Afterwards, the algorithm is extended to the biconnected case using *SPQR-trees*. An SPQR-tree $\mathcal{T}$ reflects the decomposition of a biconnected graph $G = (V, E)$ into its *triconnected components* and their relationships [5, 8]. In fact, every triconnected component $G_\mu = (V_\mu, E_\mu)$ is represented by a tree node μ in $\mathcal{T}$ where G_μ itself is called the *skeleton* of μ. The interrelationship between two triconnected components is described by a pair of so called *virtual edges*. Both virtual edges share the same endpoints that correspond to a *split pair* $\{s, t\}$. A split pair $\{s, t\}$ is either a pair of adjacent nodes in G or a *separation pair*, i.e., the removal of $\{s, t\}$ disconnects G. Every G_μ can be categorized to be one of four types based on its structure. A bundle of at least three parallel edges is referred to as *P-node*. In case G_μ is a simple cycle of length at least three, it classifies as an *S-node*, whereas if the skeleton is a simple triconnected graph, we call it an *R-node*. The leaves of $\mathcal{T}$ are formed by *Q-nodes* that are bundles of two edges, one being a virtual edge while the other corresponds to an edge of G. Usually it is convenient to root $\mathcal{T}$, hence, inducing a hierarchy on the triconnected components. Except for the root, every skeleton G_μ contains then a virtual edge $(s, t) \in E_\mu$ that represents a link to μ's parent. We refer to (s, t) as the *reference edge* of μ and to its endpoints $\{s, t\}$ as the *poles* of μ. When considering a node μ in a rooted SPQR-tree $\mathcal{T}$, μ induces a subgraph of G referred to as the *pertinent graph* of μ.

Figure 1. (a) Example in which seven successors of a vertex v are placed in a non-bitonic manner. The last three edges to be attached to v (dashed) are separated by previously attached ones (solid). In (b), when using a bitonic ordering, they appear consecutively in the embedding around v.

The main task, when extending a triconnected drawing procedure to a biconnected one using SPQR-trees, can be sketched as follows. The original algorithm serves as a basis for the case in which μ is an R-node. It is then modified such that each (virtual) edge in the drawing can be replaced recursively by a drawing of the corresponding pertinent graph. Usually a drawing has to match certain invariant properties. For S- and P-nodes alternative methods are used. Finding a good invariant and presenting a clear proof can be tedious work and its complexity may outweigh the description of the original triconnected algorithm. We offer a different approach by defining a new type of *st*-ordering whose successor lists have the aforementioned property of being bitonic.

3 The bitonic *st*-ordering

A sequence is said to be *bitonic*, if it can be partitioned into two subsequences such that one is monotonically increasing while the other is decreasing. More specifically:

▶ **Definition 3.** *An ordered sequence $A = \{a_1, \dots, a_n\}$ is **bitonic increasing**, if there exists $1 \le k \le n$ such that $a_1 \le \cdots \le a_k \ge \cdots \ge a_n$ holds and **bitonic decreasing** if $a_1 \ge \cdots \ge a_k \le \cdots \le a_n$. Moreover, we say A is bitonic increasing (decreasing) with respect to a function f if $A' = \{f(a_1), \dots, f(a_n)\}$ is bitonic increasing (decreasing).*

One property of bitonic sequences that is very useful in our context is the following:

▶ **Property 4.** *If a sequence $A = \{a_1, \dots, a_n\}$ is bitonic increasing (decreasing), then the reversed sequence $A' = \{a_n, \dots, a_1\}$ is bitonic increasing (decreasing) as well.*

In the following, we restrict ourselves to bitonic increasing sequences. Thus, we abbreviate it by just referring to it as being bitonic.

▶ **Definition 5.** *Let $G = (V, E)$ be a biconnected planar graph with a fixed embedding and $(s, t) \in E$. We say an st-ordering π is a **bitonic st-ordering**, if at every vertex $v \in V$ the ordered sequence of successors $S(v) = \{w_1, \dots, w_m\}$ as implied by the embedding is bitonic with respect to π.*

An ordering with this additional property is particularly useful in an incremental algorithm; the edges that correspond to those successors of a vertex v that have not been placed yet, appear consecutively in the embedding around v. See Figure 1 for an example. Next, we describe how to obtain such a bitonic *st*-ordering.

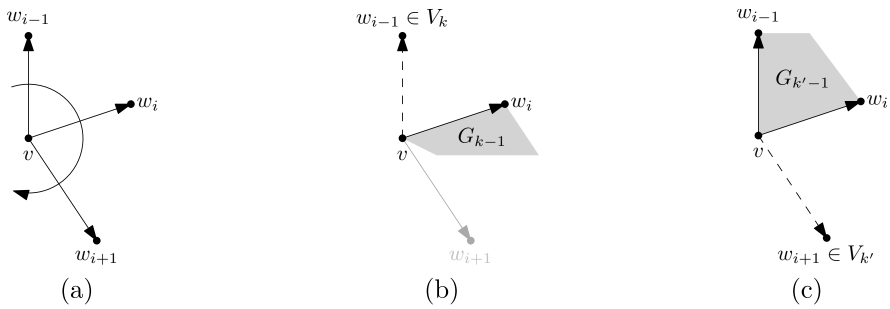


Figure 2. (a) The initial situation at v with $S(v) = \{\ldots, w_{i-1}, w_i, w_{i+1}, \ldots\}$. (b) G_{k-1} with $k = \pi'(w_{i-1})$ where w_{i-1} has to be in the outer face of G_{k-1}. (c) $G_{k'-1}$ with $k < k' = \pi'(w_{i-1})$ where w_{i+1} has to be in the outer face of $G_{k'-1}$.

▶ **Lemma 6.** *Every triconnected planar graph $G = (V, E)$ admits a bitonic st-ordering for every $(s, t) \in E$.*

Proof. From its definition it is easy to see that a canonical ordering $V_1 \cup \cdots \cup V_K$ can be transformed into an *st*-ordering π. We start by describing the construction of π and then show that it is indeed bitonic with respect to π. Given an edge $(s, t) \in E$, we compute a canonical ordering $V_1 \cup \cdots \cup V_K$ of G by choosing $V_1 = \{s, s'\}$ and $V_K = \{t\}$ with s' being the vertex that precedes t in the clockwise order around s. Notice that by definition of the canonical ordering, the edges (s, t) and (s, s') are on the outer face. For the *st*-ordering π we follow a simple principle that is sometimes referred to as the vertex ordering of a canonical ordering: Regardless of $V_k = \{v_1, \ldots, v_m\}$ with $1 \leq k \leq K$ being a chain or singleton, we choose π for $1 \leq i \leq m$ such that $\pi(v_i) = |V_1| + \cdots + |V_{k-1}| + i$.

For the sake of notation we may refer to the partition of a vertex $v \in V_k$ with $\pi'(v) = k$. Notice that by construction of π for all $u, v \in V$ with $\pi'(u) < \pi'(v)$, it holds that $\pi(u) < \pi(v)$. By definition of the canonical ordering, every $v \in V_k$ with $k < K$ has at least one neighbor w in $V_{k+1} \cup \cdots \cup V_K$. It holds then that $\pi(w) > \pi(v)$ and as a result every $v \neq t$ has at least one successor. In case $V_k = \{v\}$ $(1 < k \leq K)$ is a singleton, v has at least two neighbors, say c_l and c_r, in $V_1 \cup \cdots \cup V_{k-1}$ with $\pi(c_l) < \pi(v)$ and $\pi(c_r) < \pi(v)$, thus v has at least two predecessors. The other case, i.e., $V_k = \{v_1, \ldots, v_m\}$ $(k > 1)$ is a chain, only v_1 and v_m have one neighbor each, let us say c_l and c_r, in $V_1 \cup \cdots \cup V_{k-1}$. However, for every $v_i \in V_k$ with $i > 1$ it holds that $\pi(v_{i-1}) < \pi(v_i)$. Hence, every v_i with $i < m$ has exactly one predecessor while v_m has even two. Special attention must be paid to $V_1 = \{s, s'\}$ since for this chain no c_l and c_r exist. However, the predecessor of s' is s and s itself does not require a predecessor for π being an *st*-ordering. Since all vertices $v \neq s$ have predecessors the order in $S(v)$ is well-defined by considering them clockwise. For s we have to break the cyclic order and set $S(s) = \{t = w_1, w_2, \ldots, w_{m-1}, w_m = s'\}$.

In order to prove that π is a bitonic *st*-ordering, we first show that every successor list obtained from π is bitonic with respect to π' instead of π. To do so, assume to the contrary that there exists a successor list $S(v) = \{w_1, \ldots, w_i, \ldots, w_m\}$ of some vertex v that is not bitonic with respect to π', i.e., there is a $w_i \in S(v)$ with $1 < i < m$ for which $\pi'(w_{i-1}) > \pi'(w_i)$ and $\pi'(w_{i+1}) > \pi'(w_i)$ holds. Furthermore, let w.l.o.g. $\pi'(w_{i-1}) < \pi'(w_{i+1})$. Notice that by construction of π and $S(v)$, it follows that $\pi'(w_{i-1}) \neq \pi'(w_{i+1})$. See Figure 2a for the initial situation at v. Now we set $k = \pi'(w_{i-1})$ and $k' = \pi'(w_{i+1})$ and argue that in a canonical ordering this can only occur for $k = 2$. By definition of the canonical ordering, $w_{i-1} \in V_k$ has to be in the outer face of G_{k-1} as displayed in Figure 2b. Similarly, $w_{i+1} \in V_{k'}$ has to be in the outer face of $G_{k'-1}$ (see Figure 2c). As a result, the outer face of G_{k-1} must be on both sides of the edge (v, w_i) and there is only one such G_{k-1} for which this is the case, namely G_1. Hence, $k = 2$, $v = s$, $w_i = s'$ and $w_{i+1} = t$. However, we defined $S(s)$ such that it ends with $w_m = s'$ which is a contradiction.

It remains to show that all $S(v)$ are not only bitonic with respect to π', but also for π. As aforementioned, by construction of π from π', for two vertices $u, v \in V$ with $\pi'(u) < \pi'(v)$ it follows that $\pi(u) < \pi(v)$. And since we have just shown for the successor list $S(v) = \{w_1, \ldots, w_i, \ldots, w_m\}$ of every vertex $v \in V$ it holds that $\pi'(w_{i-1}) < \pi'(w_i)$ or $\pi'(w_{i+1}) < \pi'(w_i)$, we may deduce that $\pi(w_{i-1}) < \pi(w_i)$ or $\pi(w_{i+1}) < \pi(w_i)$. Hence, every $S(v)$ is bitonic with respect to π. ◻

The proof is constructive and reveals one additional property: The successor list of s is a special case, because it contains s' and t. Furthermore, s is the only vertex with $\pi(s) < \pi(s')$ and for every vertex $v \in V$ with $v \neq t$, $\pi(v) < \pi(t)$ holds. Since the successor list of s starts with t, ends with s' by our construction, and is bitonic with respect to π, we can state the following:

▶ **Corollary 7.** *The successor list of s starts with t, ends with s' and is sorted decreasingly with respect to π, i.e., $S(s) = \{t, w_2, \ldots, w_{m-1}, s'\}$ such that $\pi(t) > \pi(w_2) > \cdots > \pi(w_{m-1}) > \pi(s')$.*

While the above results follow the intuition of canonical orderings, they hold only for the case where the input is triconnected. Next, we extend this result to the biconnected case using SPQR-trees. Corollary 7 provides us with the necessary ingredient for an invariant. More details are given in the proof of the main result of this section:

▶ **Theorem 8.** *Every biconnected planar graph $G = (V, E)$ has a bitonic st-ordering π for any given st-edge $e^* \in E$. The ordering π and a corresponding embedding can be computed in time $\mathcal{O}(|V|)$.*

Proof. The overall challenge is to recursively compose a bitonic *st*-ordering along an SPQR-tree. For a subtree, we assume that we have already constructed a bitonic *st*-ordering that complies with an invariant. Then we show that we can combine it in the skeleton of the parent node with the solutions of other subtrees.

Invariant: For the assignment of an index in π, we maintain a single global counter that we use to label the vertices in an incremental manner. The poles $\{s, t\}$ of a tree node μ are labeled by the parent. Moreover, s has already been labeled such that we may assume that the global counter has a value greater than $\pi(s)$. Furthermore, π is a bitonic *st*-ordering for the subgraph induced by μ when assigning t the current value of the counter. Additionally, the successor list of s is sorted decreasingly with respect to π. We start by embedding G, creating the SPQR-tree $\mathcal{T}$ and rooting it at the Q-node representing the given *st*-edge $e^* = (s^*, t^*)$. Then we initialize the global counter, label s^*, and recurse on the only child of the root. Following standard practice, we now distinguish the different types of tree nodes.

Serial case: Let the skeleton of the S-node μ be the simple cycle $s, v_1, \ldots, v_{m-1}, t, s$, where (s,t) is the reference edge representing the parent of μ. The remaining edges $(s, v_1), \ldots, (v_{m-1}, t)$ correspond to the children $\mu_1, \ldots, \mu_m$ of μ. We recurse on μ_1, label v_1, recurse on μ_2, and so on, until μ_m. Notice that we do not label t. Clearly, the result is an *st*-ordering when assigning t the current value of the counter. The successor lists of $s, v_1, \ldots, v_{m-1}$ are all sorted decreasingly due to our invariant, thus, are bitonic.

Parallel case: We first check if one of the children $\mu_1, \ldots, \mu_m$ of the P-node μ is a Q-node. In that case we change the order of the children such that μ_1 is the Q-node. Notice that this implies a change in the embedding of G. Then we recurse on the children in their reverse order, i.e. $\mu_m, \ldots, \mu_1$. Consider now the successor list $S(s)$ of s: The neighbors $w^i_1, \ldots, w^i_{k'}$ with $1 \leq i \leq m$ that are located in the induced subgraph of μ_i form a consecutive sequence in $S(s)$:

$$S(s) = \{\ldots, \underbrace{w^i_1, \ldots, w^i_k}_{\text{neighbors in } \mu_i}, \underbrace{w^{i+1}_1, \ldots, w^{i+1}_{k'}}_{\text{neighbors in } \mu_{i+1}}, \ldots\}$$

By our invariant, it follows that $\pi(w^i_j) > \pi(w^i_{j+1})$ and since we recursed on $\mu_1, \ldots, \mu_m$ in reverse order, $\pi(w^i_k) > \pi(w^{i+1}_1)$ holds. Hence, the sequence is decreasing.

Rigid case: We start by constructing a temporary ordering π' for the triconnected skeleton $G_\mu = (V_\mu, E_\mu)$ of the R-node μ using Lemma 6 and choosing the reference edge (s,t) as input. Then we traverse the vertices of V_μ in the ordering as given by π'. At a vertex $v \in V_\mu$, we recurse on the incident edges $(u,v) \in E_\mu$ with $\pi'(u) < \pi'(v)$, i.e., the incoming edges of v with respect to π'. Afterwards, we label v unless $v = t$. The resulting ordering is not necessarily a bitonic *st*-ordering. We proceed in two steps: First we derive some useful properties of π and narrow down the problem. Then we argue that mirroring the embedding of some children of μ changes the successor lists such that they become bitonic with respect to π.

Let us take a closer look at the properties of π: Since we labeled all $v \in V_\mu$ in the order as provided by π', for any two vertices $u, v \in V_\mu$ with $u \neq v$, it holds that $\pi'(u) < \pi'(v)$ if and only if $\pi(u) < \pi(v)$. Hence, π is a feasible bitonic *st*-ordering for G_μ. Recall that we recursed on the children in a special way. Consider a vertex v' in the induced subgraph of a child μ_{uv} represented by the virtual edge $(u,v) \in E_\mu$ with $\pi(u) < \pi(v)$. Furthermore, assume that v' is not a pole of μ_{uv}, i.e., $u \neq v' \neq v$. Then v' has been labeled before v and after any $w \in V_\mu$ with $\pi(w) < \pi(v)$, thus $\pi(w) < \pi(v') < \pi(v)$. When now considering a fourth vertex, say w', that is defined similar as v', i.e., a non-pole vertex located in the subgraph induced by a virtual edge $(x,w) \in E_\mu$ with $\pi(x) < \pi(w)$, then we may deduce the implication $\pi(w) < \pi(v) \Rightarrow \pi(w') < \pi(v')$. Stemming from the special traversal of the edges, this property is of particular interest when considering the successor lists.

Let $S'(v) = \{w'_1, \ldots, w'_h, \ldots, w'_m\} \subset V_\mu$ be the successor list of $v \in V_\mu$. See Figure 3a for an example. Notice that $\pi(w'_1) < \cdots < \pi(w'_h) > \cdots > \pi(w'_m)$ holds. Furthermore, let $\mu_1, \ldots, \mu_m$ be the corresponding children of μ that are represented by the virtual edges $(v, w'_1), \ldots, (v, w'_m)$ with $\pi(v) < \pi(w'_i)$ for $1 \leq i \leq m$. Similar to the P-node case, we refer to the neighbors of v that are contained in the subgraph induced by μ_i

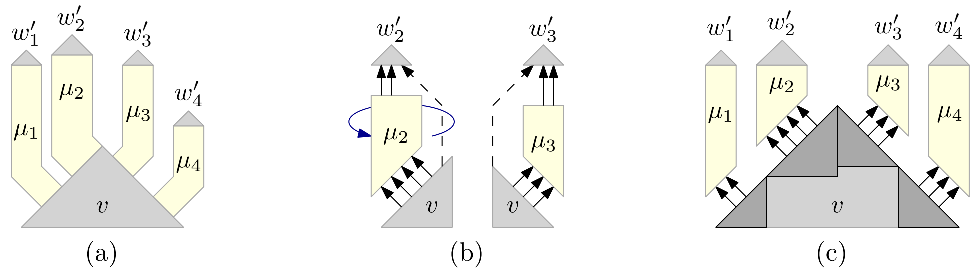


Figure 3. (a) Example of virtual edges $(v, w_1'), \ldots, (v, w_4')$ in an R-node representing the tree nodes $\mu_1, \ldots, \mu_4$. (b) Mirroring the embedding of the subgraph induced by μ_2 turning the decreasing sequence into an increasing sequence. (c) The bitonic successor list at v after mirroring the embedding of μ_1 and μ_2.

as $w_1^i, \ldots, w_{k_i}^i$. These form a consecutive sequence in $S(v)$, hence, we may write $S(v)$ as

$$S(v) = \{\underbrace{w_1^1, \ldots, w_{k_1}^1}_{\text{neighbors in } \mu_1}, \ \ldots, \ \underbrace{w_1^h, \ldots, w_{k_h}^h}_{\text{neighbors in } \mu_h}, \ \ldots, \ \underbrace{w_1^m, \ldots, w_{k_m}^m}_{\text{neighbors in } \mu_m}\}.$$

The idea now is to distinguish between two cases, depending on if either $i < h$ or $i \geq h$ holds, i.e., w_i' is in either the increasing or decreasing partition of $S'(v)$.

Let us first consider the case $h \leq i$: Since $\pi(w_i') > \pi(w_{i+1}')$ for $h \leq i < m$, it follows that $\pi(w_{k_i}^i) > \pi(w_1^{i+1})$ for all $h \leq i < m$, i.e., the last neighbor in the subgraph induced by μ_i has a greater label than the one in μ_{i+1}. By our invariant we may assume that $\pi(w_1^i) > \cdots > \pi(w_{k_i}^i)$ for all $h \leq i \leq m$ holds, i.e., with respect to π, we have a decreasing subsequence in $S(v)$. Hence, the sequence $w_1^h, \ldots, w_{k_m}^m$ is decreasing with respect to π.

In the second case where $1 \leq i < h$ holds, an increasing sequence is required. We mirror the embedding of every subgraph induced by μ_i with $1 \leq i < h$ along its poles (v, w_i'). As a result the decreasing subsequences in $S(v)$ turn into increasing ones, i.e., $\pi(w_1^i) < \cdots < \pi(w_{k_i}^i)$ for all $1 \leq i < h$ (μ_2 in Figure 3b). Notice that by Property 4 the successor list of every vertex in the mirrored subgraph remains bitonic. Now similar to the first case, we argue that from $\pi(w_i') < \pi(w_{i+1}')$ it follows that $\pi(w_{k_i}^i) < \pi(w_1^{i+1})$ for all $1 \leq i < h$. Thus, the sequence $w_1^1, \ldots, w_{k_{h-1}}^{h-1}$ is increasing with respect to π. And as a result, the sequence $w_1^1, \ldots, w_{k_{h-1}}^{h-1}, w_1^h, \ldots, w_{k_m}^m$ is bitonic with respect to π (Figure 3c). Notice that for $v = s$, there exists no i with $\pi(w_i') < \pi(w_{i+1}')$, thus, $S(s)$ is sorted decreasingly with respect to π as required by the invariant.

The case where μ is a Q-node is trivial. Both, the canonical ordering and the SPQR-tree, can be computed in linear time, thus, the runtime follows immediately. $\square$

In the proof of the main theorem, we changed the embedding of G in two places. At first in the P-node case, we had to ensure that a possible Q-node follows the reference edge in clockwise order around s. Afterwards in the R-node case, we mirrored the embedding along the poles to turn a decreasing sequence into an increasing one. The latter change is caused by our invariant that only provides a decreasing sequence at s for the sake of an easier maintainable invariant. In an actual implementation, this

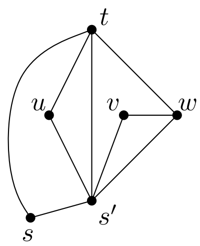


Figure 4. A graph for which no bitonic *st*-ordering exists for the given embedding.

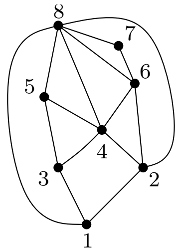


Figure 5. Running example with its bitonic *st*-ordering and corresponding embedding.

can easily be avoided by mirroring the embedding twice, once before recursing on the corresponding child and then afterwards. Thus, the resulting embedding is equivalent to the initial one. However, for the P-node case it is not trivial and the question may arise if it is necessary in general, or if one may always find a bitonic *st*-ordering for every edge when a fixed embedding is given. To answer this question, we give a small counterexample.

▶ **Lemma 9.** *Given a fixed embedding, there exist biconnected planar graphs that do not admit a bitonic st-ordering for every edge.*

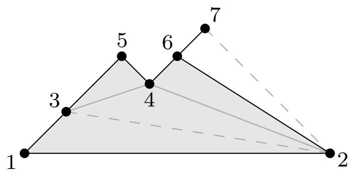
Proof. Consider the graph in Figure 4 and its embedding. The triangle consisting of s',t and w is attached to the source s via s'. Clearly, in any feasible *st*-ordering $\pi(u) < \pi(t)$ and $\pi(v) < \pi(w) < \pi(t)$ must hold. Thus, the successor list $S(s') = \{u, t, v, w\}$ of s' as implied by the illustrated embedding is not bitonic with respect to π, because it follows that $\pi(u) < \pi(t) > \pi(v) < \pi(w)$, which is neither bitonic increasing nor decreasing. ☐

Although this is a drawback, it is worth mentioning that in many approaches that employ SPQR-trees for drawing purposes, implicit changes to the embedding are made anyway.

4 Applications

In the following, we present two simple applications of bitonic *st*-orderings. The results are not new, but we believe that the bitonic *st*-ordering simplify things. By its nature, it works out of the box for biconnected planar graphs and therefore no augmentation of the input is required. For both applications, we assume that a biconnected planar graph $G = (V, E)$ with a bitonic *st*-ordering π and the corresponding embedding is given. The graph, its embedding and ordering displayed in Figure 5 serves as a running example.

We start with a classic problem: Straight-line drawings of biconnected graphs by borrowing some ideas from Harel and Sardas [9]. They first describe an algorithm to obtain a biconnected canonical ordering. Then a modification of the classic algorithm of de Fraysseix, Pach and Pollack [3] is used to obtain a planar straight-line layout. We only outline the approach here: during every step k, the algorithm maintains a straight-line drawing for the already placed vertices, $v_1, \ldots, v_{k-1}$ of the biconnected canonical ordering. Similar to the original algorithm, they maintain for the contour of



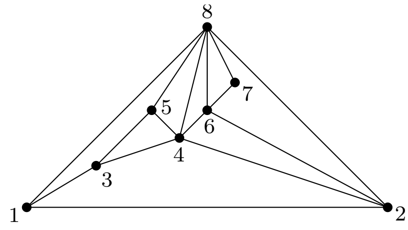


Figure 6. (a) Adding v_7 which has only one predecessor but right support. (b) The final straight-line drawing produced by the modified de Fraysseix-Pach-Pollack algorithm.

the outer face of G_{k-1} the property that it consists only of segments with slopes $+1$ or -1. Adding a new vertex v_k with leftmost neighbor c_l and rightmost neighbor c_r in G_{k-1} results in a stretch of the drawing such that the edges (v_k, c_l) and (v_k, c_r) have slope $+1$ and -1, respectively. Of course, this works only for $c_l \neq c_r$. In the other case, where $c_l = c_r$ holds, i.e., v_k has only one predecessor, say $u = c_l = c_r$, one has to decide if v_k is placed to the right or to the left of u. Harel and Sardas [9] introduce for those vertices the property of having *left* or *right support*. Their ordering guarantees that either the successor or predecessor of v_k in the clockwise ordering around u has already been placed. Since π has by definition the same property, we may proceed similar. Avoiding sub cases, we always try to place v_k to the left, i.e., choosing a new c_l such that c_l is the predecessor of $u = c_r$ on the contour of G_{k-1}. However, in the case where there exists a w that precedes v_k in $S(u)$ and for which $\pi(w) > \pi(v_k)$ holds, we have to place v_k to the right by choosing $c_l = u$ and c_r to be the successor of u on the contour. Figure 6b shows an example generated by our implementation. Notice that in difference to the ordering as proposed in [9], in an *st*-ordering every vertex except of t has a successor, hence the faces of the drawing are y-monotone.

Next, we turn our attention to the second application: *contact representations* using rectilinear T-shaped polygons. Alam et al. [1] recently used these as an intermediate step to create cartograms. The idea is to represent a planar graph by touching sides of simple interior-disjoint polygons, in this case upside-down oriented T-shaped polygons. Their approach employs *Schnyder realizer* and their close relationship to canonical orderings. For more details see [1]. However, we choose a different approach and consider instead a special *visibility representation* of G. We assume that the reader is familiar with the basics of visibility representations. For an introduction, see e.g. [4]. The common way to obtain such a visibility representation can be summarized as follows: The y-coordinates $y(v)$ of the horizontal segments that represent the vertices $v \in V$ of G are computed by an optimal topological ordering of a planar *st*-graph induced by an *st*-ordering. For the x-coordinate $x(e)$ of a vertical segment that represents an edge $e \in E$, the same procedure is repeated but on the dual planar *st*-graph. We skip the first step and choose π itself for the y-coordinates, i.e., $y(v) = \pi(v)$. As a result every vertex has now its own row that corresponds to its rank in π. See Figure 7c for such a visibility representation for the running example. Although a visibility representation can be derived this way for any *st*-ordering, we may now benefit from the property that π is a bitonic *st*-order. Since for every $v \in V$, $S(v)$ is bitonic with respect to π, by construction it is also bitonic with respect to the y-coordinates, i.e., the successors are

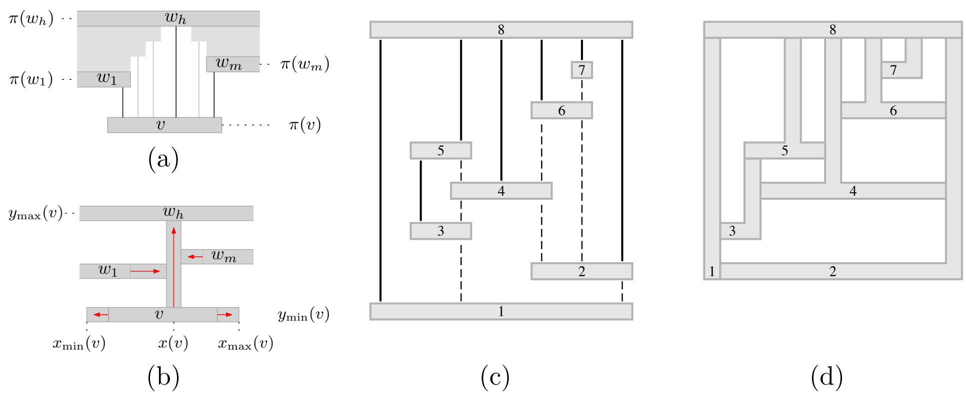


Figure 7. (a) Successors $w_1, \ldots, w_h, \ldots, w_m$ of v whose ordering in the embedding is bitonic with respect to the y-coordinates. (b) Creating a pole at w_h, i.e., the highest successor, and pulling the bars of the remaining towards it. (c) Visibility representation for the running example. The edges to the highest successor are drawn solid. (d) The resulting T-shaped contact representation.

located above v in an increasing and then a decreasing staircase pattern. See Figure 7a for an illustration.

By using a simple trick, we now transform this wedge-like structure into a rectilinear T-shaped polygon. The idea is straightforward: We create a vertical segment on top of the horizontal bar that reaches all the way up to w_h, i.e., the highest successor of v. Afterwards we pull the bars of the remaining successors towards this pole. See the arrows in Figure 7b for a sketch of the idea. Notice that in case of a non-bitonic *st*-ordering, a single pole is not sufficient. More specifically, let $x_{\min}(v)$ ($x_{\max}(v)$) denote the left (right) border of the upside-down T representing v and $x(v)$ the horizontal offset of the pole. Furthermore, let $y_{\min}(v)$ and $y_{\max}(v)$ denote the vertical offset of the horizontal bar and the upper border of the pole, respectively. Then, for every $v \in V$ with $S(v) = \{w_1, \ldots, w_h, \ldots, w_m\}$ in which $y(w_1) < \cdots < y(w_h) > \cdots > y(w_m)$ holds, we create the vertical segment by choosing $x(v) = x((v, w_h))$, where $x(v, w_h)$ denotes the x-coordinate of (v, w_h) in the visibility representation. Furthermore, we set $y_{\max}(v) = y_{\min}(w_h)$. For the remaining successors w_i with $1 \leq i < h$, i.e., those located to the left of the pole, we establish contact with the pole from the left by choosing $x_{\max}(w_i) = x(v)$. In a symmetric manner, we set $x_{\min}(w_i) = x(v)$ with $h < i \leq m$ for those successors that are located on the right. Notice that $x_{\min}(v)$ and $x_{\max}(v)$ are only defined in the case where there exists such a pole on both sides. Otherwise, we have to ensure that the horizontal bar of v covers at least the attaching poles from below. Hence, for every u with $v \in S(u)$ and $\pi(v) = \max_{w \in S(u)}\{\pi(w)\}$, i.e., all u for which v is the highest successor, we set $x_{\max}(v) = \max\{x(v), x(u)\}$ and $x_{\min}(v) = \min\{x(v), x(u)\}$. See v_3 and v_5 in Figure 7c. The final contact representation for our running example is shown in Figure 7d.

A larger example for both applications with $|V| = 80$ and $|E| = 150$ can be found in the appendix. In Figure 8 the visibility and corresponding contact representation is displayed. The straight-line drawing of the graph is shown in Figure 9.

5 Implementation details

The presented work has been implemented in C++ using the *Open Graph Drawing Framework (OGDF)* [11]. For the canonical ordering, we implemented the *leftist canonical ordering* algorithm as described by Badent et al. [2]. The linear-time implementation of Gutwenger and Mutzel [8] is used for the SPQR-tree that is required for Theorem 8. It is part of the OGDF, publicly available and provides a convenient interface to navigate the tree and the skeletons.

6 Conclusion

We have shown that every biconnected planar graph has a bitonic *st*-order that can be obtained in linear time. Moreover, two applications have been presented, both requiring the property of being bitonic. We believe that the bitonic *st*-ordering is a useful addition to the set of existing tools. Besides having potentially a broad range of applications, it may simplify existing methods considerably.

References

[1] M. J. Alam, T. Biedl, S. Felsner, M. Kaufmann, S. G. Kobourov, and T. Ueckerdt. Computing cartograms with optimal complexity. In *Proceedings of the Twenty-eighth Annual Symposium on Computational Geometry*, SoCG '12, pages 21–30. ACM, 2012.

[2] M. Badent, U. Brandes, and S. Cornelsen. More canonical ordering. *Journal of Graph Algorithms and Applications*, 15(1):97–126, 2011.

[3] H. de Fraysseix, J. Pach, and R. Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990.

[4] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, Englewood Cliffs, NJ, 1999.

[5] G. Di Battista and R. Tamassia. Incremental planarity testing. In *30th Annual Symposium on Foundations of Computer Science, 1989*, pages 436–441, 1989.

[6] S. Even and R. E. Tarjan. Computing an st-numbering. *Theoretical Computer Science*, 2(3):339 – 344, 1976.

[7] C. Gutwenger and P. Mutzel. Planar polyline drawings with good angular resolution. In S. Whitesides, editor, *Graph Drawing*, volume 1547 of *Lecture Notes in Computer Science*, pages 167–182. Springer Berlin Heidelberg, 1998.

[8] C. Gutwenger and P. Mutzel. A linear time implementation of SPQR-trees. In J. Marks, editor, *Graph Drawing*, volume 1984 of *Lecture Notes in Computer Science*, pages 77–90. Springer Berlin Heidelberg, 2001.

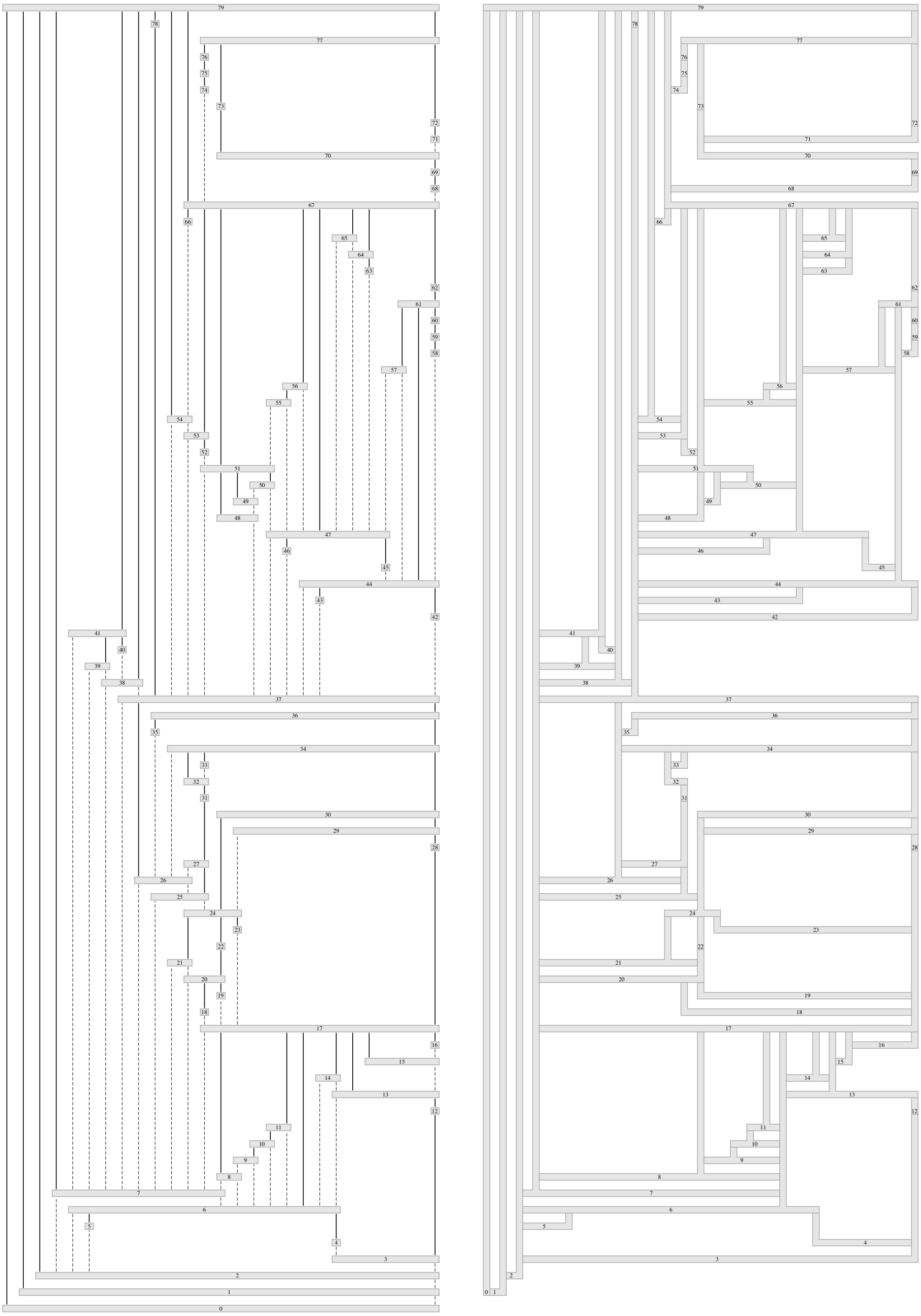
[9] D. Harel and M. Sardas. An algorithm for straight-line drawing of planar graphs. *Algorithmica*, 20(2):119–135, 1998.

[10] G. Kant. Drawing planar graphs using the canonical ordering. *Algorithmica*, 16:4–32, 1996.

[11] OGDF - Open Graph Drawing Framework. http://www.ogdf.net/.

[12] R. Tamassia. *Handbook of Graph Drawing and Visualization (Discrete Mathematics and Its Applications)*. Chapman & Hall/CRC, 2007.

Appendix



(a) (b)

Figure 8. (a) Larger example ($|V| = 80$, $|E| = 150$) of a visibility representation obtained from a bitonic st-ordering and (b) the corresponding contact representation.

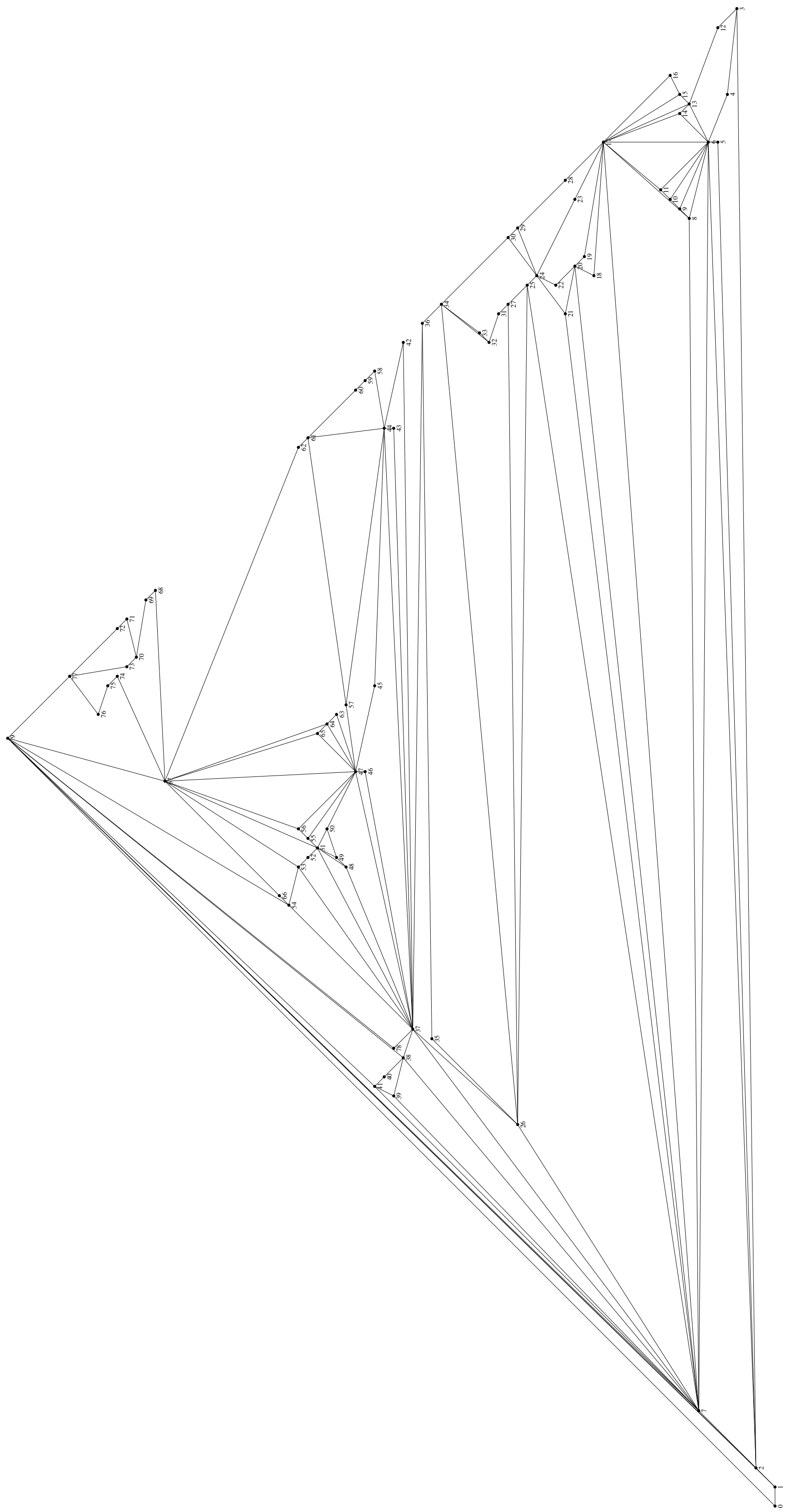


Figure 9. The output of the modified FPP algorithm for the larger example.